EMPIRICAL SCHEDULING OF NETWORK PACKETS

BACKGROUND OF THE INVENTION

[01] The present invention relates generally to a system for allowing devices connected to a network (e.g., an IP or Ethernet network) to collaborate with other such devices so as to transmit and receive data packets without impairment on the network

[02] As is generally known, Ethernet and Internet Protocol (IP) are systems for transmitting packets between different points on a communications network. These switching systems are known as "contention-based" systems. That is, all transmitters contend for network resources. All transmitters may transmit simultaneously. If they do, then network resources may be oversubscribed. When this happens, data may be delayed or lost, resulting in network impairment.

[03] As illustrated in FIG. 1, four streams of packets are input to a packet switch 112, which routes the packets to one or more outputs based on addressing information contained in each packet. Packets may arrive at the switch at unpredictable times, leading to bursts of inputs that must be handled. The switch typically maintains a packet queue 114 that is able to store a small number of packets. The queue may comprise multiple queues arranged by packet priority level, such that priority 3 packets, for example, take precedence over priority 1 packets. If the inputs are too bursty, the queues fill up and some packets may be discarded. The higher-priority queues are typically emptied before the lower-priority queues, such that the lower-priority queues are more likely to lose data first.

[04] IP systems suffer from impairments such as packet loss and jitter. This happens because there is no control over how many such packets reach a router at any given instant. If two packets arrive at a router at the same time, destined for the same port, one will have to be delayed. Both cannot be transmitted simultaneously. One of the packets will be saved in the queue until the first packet is completely transmitted.

[05] FIG. 2 shows a computer network comprising endpoints 100, 101, 102, and 103. The network includes routers 104 through 107. As can be seen in the figure, if endpoints 100 and 101 communicate with endpoints 102 and 103 at the same time, a bottleneck may develop between routers 105 and 106. This may occur because too many packets may be simultaneously transmitted between the routers, causing the routers to discard overflow packets. This can happen even at low levels of network utilization.

[06] Various methods have been developed to overcome data loss on Ethernet and IP networks. The primary approach has been to use additional protocols to replace lost data. This is an after-the-fact solution. An example is the well-known Transmission Control Protocol (TCP). TCP is able to detect data loss and it causes retransmission of the data, until a perfect copy of the complete data file is delivered to the recipient device.

[07] Many devices may be unable to use TCP or any retransmission method because it is far too slow. Real-time applications require delivery of data, accurately, the first time. For these applications to operate well, even the speed of light causes undesired delay. It is not feasible or desirable to add retransmission delay.

[08] The problem is determining how to provide reliable, first-time delivery on a contention-based network. Various approaches have been tried. The most commonly proposed system relies on prioritization of data in the network. With this approach, data having real-time constraints is identified with priority coding so that it may be transmitted before other data.

[09] Prioritization seems at first to be a good solution. However, on reflection it suffers from the same difficulty. Prioritization only provides a delivery advantage relative to the lower-priority data. It provides no advantage against the other priority data. Analysis and testing shows that this approach can work in certain circumstances, but only when the amount of priority data is small. For simple applications like voice, the percentage of the total may need to be 8% or less. Other applications must occupy an even smaller percentage of total network resource. As shown in FIG. 1, even high-

priority packets may be discarded if too many high-priority packets are transmitted within a short time interval. For many networks this makes prioritization impractical.

[10] Another approach is to multiplex the data. With this method the bursts of data associated with one flow of data are separated from the burst of another. Multiplexing usually uses some type of time-domain system (known as Time Domain Multiplexing (TDM)) to separate flows. Flows may be separated in groups, so that one group does not contend with another group. This can be an improvement but still leaves the possibility of contention between groups. The only way to eliminate contention is to multiplex each flow individually. A central problem with multiplexing is that it eliminates a principal advantage of the network, namely that average bandwidth available to all is reduced. In other words, each potential transmitter on the network is guaranteed a slot of time on the network, even if that time is infrequently used. This leads to inefficient resource usage.

[11] Asynchronous Transfer Mode (ATM) is another technology for multiplexing a data network, to reduce contention. ATM breaks all data flows into equal length data blocks. Further, ATM can limit the number of data blocks available to any flow or application. The result is a virtual TDM multiplex system.

[12] Both TDM and ATM provide contention reduction, but at the cost of considerable added complexity, cost, components, and lost bandwidth performance. Other approaches rely on specialized hardware to schedule packet delivery, driving up hardware costs.

SUMMARY OF THE INVENTION

[13] The invention overcomes many of the above-identified disadvantages by providing an empirically determined delivery schedule for packets that are to be delivered between two endpoints on the network. A transmitting node having the need to transmit packets according to a known data rate (e.g., to support a voice telephone call) transmits a series of test packets over the network to the intended recipient using

different delivery times. The test packets are evaluated to determine which of the delivery times suffered the least latency and/or packet loss, and that delivery time is used to schedule the packets for the duration of the transmission. Other endpoints use a similar scheme, such that each endpoint is able to evaluate which delivery schedule is best suited for transmitting packets with the least likely packet loss and latency. Different priority levels are used to transmit the data; the test packets; and other data in the network. The system empirically determines a desirable time schedule for transmission of data packets between two endpoints on the network. The delivery scheme can be implemented without specialized hardware.

BRIEF DESCRIPTION OF THE DRAWINGS

[14] FIG. 1 shows the problem of bursty packets creating an overflow condition at a packet switch, leading to packet loss.

[15] FIG. 2 shows how network congestion can lead to a bottleneck where two sets of endpoints share a common network resource under bursty conditions.

[16] FIG. 3 shows one approach for assigning different priority levels to scheduled data (realtime level); test packets (discovery level); and other network traffic (data level).

[17] FIG. 4 shows a frame structure in which a delivery schedule can be decomposed into a master frame; subframes; and secondary subframes.

[18] FIG. 5 shows a flow chart having steps for carrying out various principles of the invention.

[19] FIG. 6 shows a system using a delivery schedule for test packets from a first endpoint to a second endpoint.

[20] FIG. 7 shows a system wherein queues for realtime traffic (priority 3) are nearly full at one packet switch and yet the traffic still gets through the network.

DETAILED DESCRIPTION OF THE INVENTION

[21] According to one variation of the invention, a priority scheme is used to assign priority levels to data packets in a network such that delivery of packets intended for real-time or near real-time delivery (e.g., phone calls, video frames, or TDM data packets converted into IP packets) are assigned the highest priority in the network. A second-highest priority level is assigned to data packets that are used for testing purposes (i.e. the so-called test packets). A third-highest priority level is assigned to remaining data packets in the system, such as TCP data used by web browsers. FIG. 3 illustrates this scheme. These priority levels can be assigned by enabling the packet priority scheme already available in many routers.

[22] Other priority levels above and below these three levels can be accommodated as well. For example, a priority level above the real-time level can be assigned for emergency purposes, or for network-level messages (e.g., messages that instruct routers or other devices to perform different functions).

[23] FIG. 4 shows how an arbitrary delivery time period of one second (a master frame) can be decomposed into subframes each of 100 millisecond duration, and how each subframe can be further decomposed into secondary subframes each of 10 millisecond duration. Each secondary subframe is in turn divided into time slots of 1 millisecond duration. According to one variation of the invention, the delivery time period for each second of transmission bandwidth is decomposed using a scheme such as that shown in FIG. 4 and packets are assigned to one or more time slots according to this schedule for purposes of transmitting test packets and for delivering data using the inventive principles. In this sense, the scheme resembles conventional TDM systems. However, unlike TDM systems, no endpoint can be guaranteed to have a particular timeslot or timeslots. Instead, nodes on the network transmit using timeslots that are empirically determined to be favorable based on the prior transmission of test packets between the two endpoints.

[24] FIG. 5 shows method steps that can be used to carry out the principles of the invention. Beginning in step 501, a determination is made that two endpoints on the network (e.g., and Ethernet network or an IP network) desire to communicate. This determination may be the result of a telephone receiver being picked up and a telephone number being dialed, indicating that two nodes need to initiate a voice-over-IP connection. Alternatively, a one-way connection may need to be established between a node that is transmitting video data and a receiving node. Each of these connection types can be expected to impose a certain amount of data packet traffic on the network. For example, a voice-over-IP connection may require 64 kilobits per second transfer rate using 80-byte packet payloads (not including packet headers). A video stream would typically impose higher bandwidth requirements on the network.

[25] Note that for two-way communication, two separate connections must be established: one for node A transmitting to node B, and another connection for node B transmitting to node A. Although the inventive principles will be described with respect to a one-way transmission, it should be understood that the same steps would be repeated at the other endpoint where a two-way connection is desired.

[26] In step 502, a delivery schedule is partitioned into time slots according to a scheme such as that illustrated in FIG. 4. (This step can be done in advance and need not be repeated every time a connection is established between two endpoints). The delivery schedule can be derived from a clock such as provided by a Global Positioning System (GPS). As one example, an arbitrary time period of one second can be established for a master frame, which can be successively decomposed into subframes and secondary subframes, wherein each subframe is composed of 10 slots each of 10 milliseconds in duration and each secondary subframe is compose of 10 slots each of 1 millisecond in duration. Therefore, a period of one second would comprise 1,000 slots of 1 millisecond duration. Other time periods could of course be used, and the invention is not intended to be limited to any particular time slot scheme.

[27] In step 503, the required bandwidth between the two endpoints is determined. For example, for a single voice-over-IP connection, a bandwidth of 64 kilobits per second

might be needed. Assuming a packet size of 80 bytes or 640 bits (ignoring packet overhead for the moment), this would mean that 100 packets per second must be transmitted, which works out to (on average) a packet every 10 milliseconds. Returning to the example shown in FIG. 4, this would mean transmitting a packet during at least one of the slots in the secondary subframe at the bottom of the figure. (Each slot corresponds to one millisecond).

[28] In step 504, a plurality of test packets are transmitted during different time slots at a rate needed to support the desired bandwidth. Each test packet is transmitted using a "discovery" level priority (see FIG. 3) that is higher than that accorded to normal data packets (e.g., TCP packets) but lower than that assigned to realtime data traffic (to be discussed below). For example, turning briefly to FIG. 6, suppose that the schedule has been partitioned into one millisecond time slots. The test packets might be transmitted during time slots 1, 3, 5, 7, 9, 11, and 12 as shown. Each test packet preferably contains the "discovery" level priority; a timestamp to indicate when the packet was sent; a unique sequence number from which the packet can be identified after it has been transmitted; and some means of identifying what time slot was used to transmit the packet. (The time slot might be inferred from the sequence number). The receiving endpoint upon receiving the test packets returns the packets to the sender, which allows the sender to (a) confirm how many of the sent packets were actually received; and (b) determine the latency of each packet. Other approaches for determining latency can of course be used. The evaluation can be done by the sender, the recipient, or a combination of the two.

[29] In step 506, the sender evaluates the test packets to determine which time slot or slots are most favorable for carrying out the connection. For example, if it is determined that packets transmitted using time slot #1 suffered a lower average dropped packet rate than the other slots, that slot would be preferred. Similarly, the time slot that resulted in the lowest packet latency (round-trip from the sender) could be preferred over other time slots that had higher latencies. The theory is that packet switches that are beginning to be stressed would have queues that are beginning to fill up, causing increases in latency and dropped packets. Accordingly, according to the inventive

principles other time slots could be used to avoid transmitting packets during periods that are likely to increase queue lengths in those switches. In one variation, the time slots can be "overstressed" to stretch the system a bit. For example, if only 80-byte packets are actually needed, 160-byte packets could be transmitted during the test phase to represent an overloaded condition. The overloaded condition might reveal bottlenecks where the normal 80-byte packets might not.

[30] Rather than the recipient sending back time-stamped packets, the recipient could instead perform statistics on collected test packets and send back a report identifying the latencies and dropped packet rates associated with each time slot.

[31] As explained above, packet header overhead has been ignored but would typically need to be included in the evaluation process (i.e., 80-byte packets would increase by the size of the packet header). Slot selection for the test packets could be determined randomly (i.e., a random selection of time slots could be selected for the test packets), or they could be determined based on previously used time slots. For example, if a transmitting node is already transmitting on time slot 3, it would know in advance that such a time slot might not be a desirable choice for a second connection. As another example, if the transmitting node is already transmitting on time slot 3, the test packets could be transmitted in a time slot that is furthest away from time slot 3, in order to spread out as much as possible the packet distribution.

[32] In step 506, a connection is established between the two endpoints and packets are transmitted using the higher "realtime" priority level and using the slot or slots that were determined to be more favorable for transmission. Because the higher priority level is used, the connections are not affected by test packets transmitted across the network, which are at a lower priority level. In one variation, the IP precedence field in IP packet headers can be used to establish the different priority levels.

[33] FIG. 6 shows a system employing various principles of the invention. As shown in FIG. 6, two endpoints each rely on a GPS receiver for accurate time clock synchronization (e.g., for timestamping and latency determination purposes). The IP

network may be comprised of a plurality of routers and/or other network devices that are able to ultimately route packets (e.g., IP or Ethernet packets) from one endpoint to the other. It is assumed that the organization configuring the network has the ability to control priority levels used on the network, in order to prevent other nodes from using the discovery priority level and realtime priority level.

[34] It should be appreciated that rather than transmitting test packets simultaneously during different time slots, a single slot can be tested, then another slot, and so on, until an appropriate slot is found for transmission. This would increase the time required to establish a connection. Also, as described above, for a two-way connection, both endpoints would carry out the steps to establish the connection.

[35] It should also be understood that the phase of all frames may be independent from one another; they need only be derived from a common clock. Different endpoints need not have frames synchronized with each other. Other approaches can of course be used.

[36] The invention will also work with "early discard" settings in router queues since the empirical method would detect that a discard condition is approaching.

[37] In another variation, packet latencies and packet dropped rates can be monitored during a connection between endpoints and, based on detecting a downward trend in either parameter, additional test packets can be transmitted to find a better time slot in which to move the connection.

[38] FIG. 7 shows a system in which a first endpoint 701 communicates with a second endpoint 706 through a plurality of packet switches 703 through 705. Each packet switch maintains a plurality of packet queues. For illustrative purposes, four different priority levels are shown, wherein 4 is the highest level, and level 1 is the lowest level. Assume that endpoint 701 attempts to initiate a connection with endpoint 706 through the network. Endpoint 701 transmits a plurality of "test" packets using

priority level 2. As can be seen, packet switch 703 is lightly loaded and the queues have no difficulty keeping up with the traffic.

[39] Packet switch 704, however, is heavily loaded. In that switch, the queue for priority level 1 traffic is full, leading to dropped packets and latencies. Similarly, the test packets transmitted by endpoint 701 at priority level 2 cause that queue to overflow, causing dropped packets and longer latencies. However, the priority level 3 queue (existing realtime traffic) is not yet full, so those packets are transported through the network unaffected. In accordance with the invention, upon detecting that test packets sent during certain time slots are dropped and/or suffer from high latencies, endpoint 701 selects those time slots having either the lowest drop rate and/or the lowest latencies, and uses those time slots to schedule the packets (which are then transmitted using level 3 priority).

[40] It is assumed that each endpoint in FIG. 7 comprises a node (i.e., a computer having a network interface) including computer-executable instructions for carrying out one or more of the above-described functions.

[41] While the invention has been described with respect to specific examples including presently preferred modes of carrying out the invention, those skilled in the art will appreciate that there are numerous variations and permutations of the above described systems and techniques that fall within the spirit and scope of the invention as set forth in the appended claims. Any of the method steps described herein can be implemented in computer software and stored on computer-readable medium for execution in a general-purpose or special-purpose computer, and such computer-readable media is included within the scope of the intended invention. Numbering associated with process steps in the claims is for convenience only and should not be read to imply any particular ordering or sequence.